VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Digital Assessment 2

M.Tech

Artificial Intelligence and Machine Learning (in collaboration with L&T)

Fall Semester 2024 - 2025

Name of Students	Patel Vishrutkumar Dineshbhai, RIshabh Chawda
Registration Number	24MAI0108, 24MAI0102
Course Name	Programming for Data Science
Course Code	MCSE627L
Faculty Name	SASIKALA .R
Assessment Number	2

Stock Market Dashboard Project Report

Project Members:

- Vishrut Patel
- Rishabh Chavda

Video Link:
Watch the Project Video Explanation:

https://youtu.be/JRfSxT3SLpc

Github Link:

https://github.com/VISHRUT225/Stock_Market_Dashboard

1. Introduction

The Stock Market Dashboard project is designed to provide users with a robust platform for analyzing stock data efficiently. This project uses real-time data from Yahoo Finance to offer an interactive interface where users can view stock price trends, trading volume, and other relevant financial metrics. Our dashboard's goal is to create an intuitive, data-rich platform for stock market insights.

2. Project Setup

Technologies Used

- **Python & Flask**: Used for backend API development.
- **yfinance**: A Python library that provides access to Yahoo Finance data, enabling us to retrieve and manage stock data.
- **JavaScript (ECharts library)**: For interactive data visualization.
- **HTML & CSS**: For creating a user-friendly and responsive interface.

Project Files

1. `app.py`: Main application file containing backend routes and functions to retrieve stock data.
2. `dashboard.html`: The main page layout, where users can select stocks and view basic information.
3. `analytics.html`: This page contains advanced data visualization, such as candlestick charts and bar charts.
4. `future_analytics.html`: A page intended for future development focused on predictive analytics.
5. `dashboard.js` and `analytics.js`: JavaScript files for handling API requests and rendering charts.
6. `style.css`: Styling for all pages to create a cohesive design and responsive layout.
7. `requirements.txt`: A file listing all necessary Python libraries, including Flask, yfinance, and pandas, to facilitate project setup on any system.

---

3. Backend Logic

Routes and Functionality in `app.py`

The backend of our application is managed by `app.py`, which includes the following key routes:

- **Dashboard Route (`/`)**: Loads the main dashboard page.
- **Stock Data API (`/get_stock_data`)**: Accepts stock symbols and timeframes as parameters to return open, close, high, and low prices, as well as trading volume and turnover.
- **Analytics Page (`/analytics`) and Future Analytics Page (`/future_analytics`)**: Loads pages dedicated to advanced analytics and future predictive functionality.

---

4. Frontend Design and User Interaction

The frontend structure includes three main pages:

1. **Dashboard Page**: The homepage allows users to select from stocks like Apple, Google, and Tesla. Key data metrics, including current, high, and low prices, are displayed, with timeframe selectors to adjust data views.
2. **Analytics Page**: The analytics page includes multiple chart types, such as:
    - **Candlestick Chart**: Provides open, close, high, and low prices over a specified timeframe.
    - **Bar Chart**: Shows yearly closing prices for the last decade.

    - **Area Chart**: Displays monthly average prices, selectable by year.
3. **Future Analytics Page**: Intended for predictive features currently under development. The goal is to use advanced analytics to forecast future stock trends.

---

5. Core JavaScript Functions

`dashboard.js`

Handles user interaction on the dashboard page:

- **`loadStockData`**: Fetches data based on selected stock symbols.
- **`renderChart`**: Dynamically updates charts based on timeframe selection.

`analytics.js`

Enables interactive data visualization in the analytics section:

- **`fetchAndRenderCandlestickData`**: Renders the candlestick chart.
- **`fetchBarChartData`**: Displays yearly closing prices over a ten-year period.
- **`fetchAreaChartData`**: Shows monthly average prices based on the selected year.

---

6. Styling and User Experience

Styling (style.css): The design is structured to be visually appealing and responsive. Key UI elements include:

- A fixed, gradient navbar for easy navigation.
- Interactive buttons and responsive layouts, ensuring usability across various devices.

---

7. Challenges and Future Improvements

During development, we faced challenges in managing real-time data updates and ensuring seamless integration between frontend and backend components. Future enhancements include adding predictive analytics, enabling users to forecast stock trends, and further improving the dashboard's interactivity.

---

8. Conclusion

The Stock Market Dashboard provides a comprehensive solution for analyzing stock data interactively. With further development, it can evolve into a powerful tool for market trend forecasting and financial analytics.

For more details, please refer to our project video: Watch the Project Video Explanation.